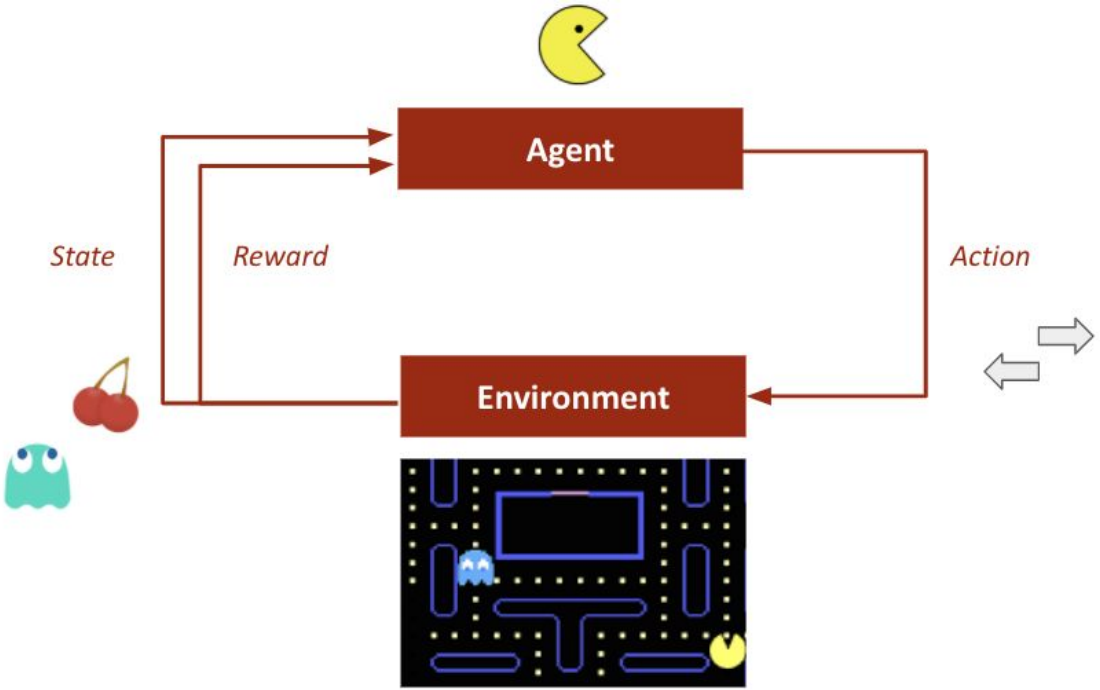
Open Problems @ BeyondRL Workshop: Reincarnating Reinforcement Learning

Rishabh Agarwal

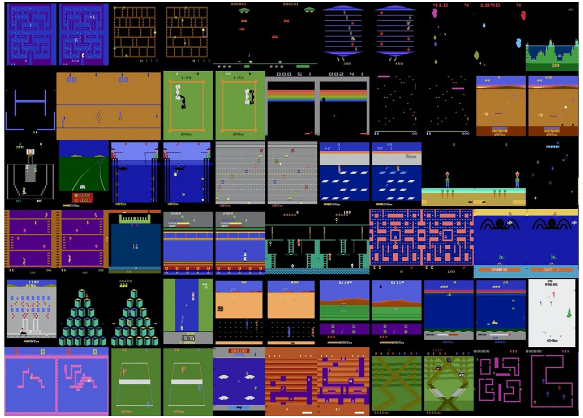
Google Research

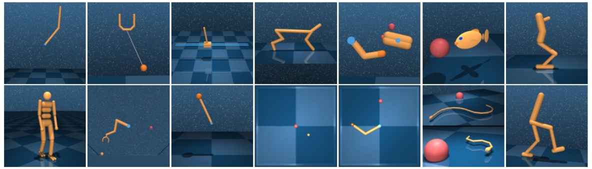
bit.ly/reincarnating_rl

Tabula rasa Reinforcement Learning

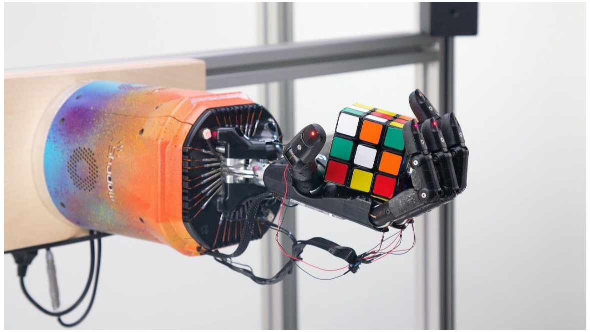


Large-scale systems: ~~Tabula rasa~~ workflow



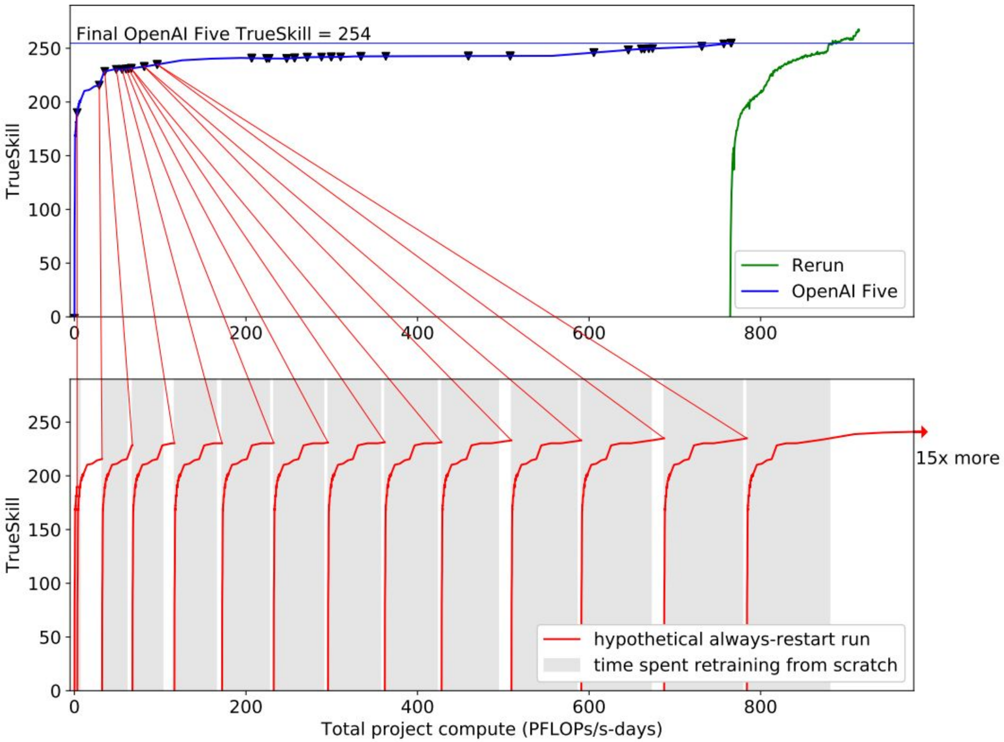


Works well here.



Not so much here.

Playing DOTA with large-scale RL training

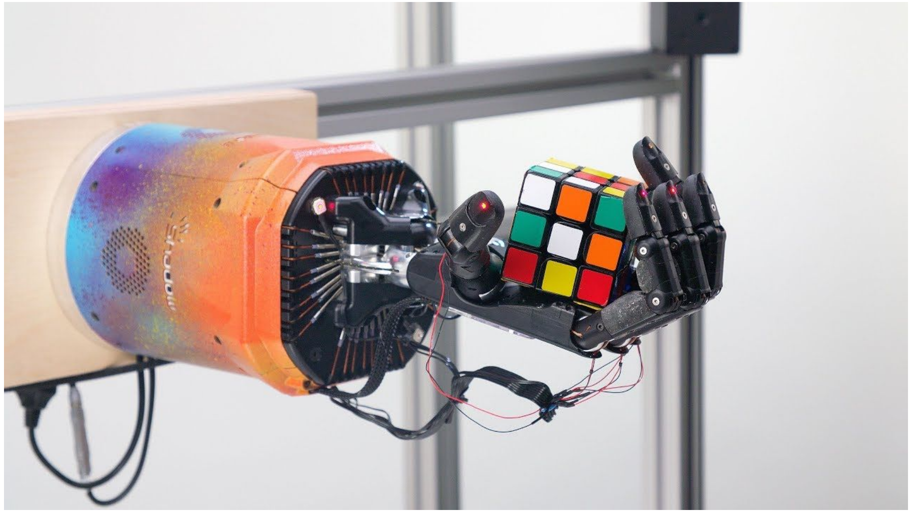


Actual learning curve (10 months)

Restarting from scratch (~40 months)

Berner, Christopher, et al. "Dota 2 with large scale deep reinforcement learning." *arXiv preprint arXiv:1912.06680* (2019).

Solving Rubik's cube with a robot hand

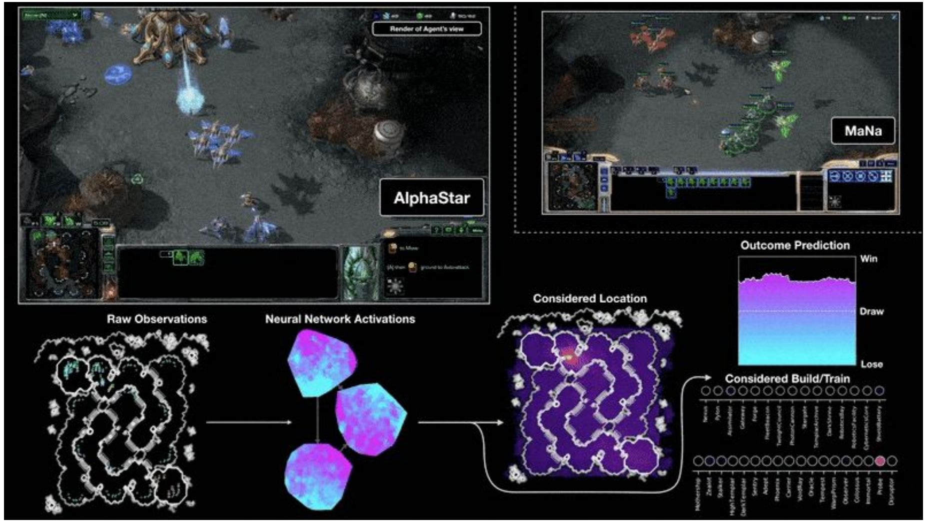


"We rarely trained experiments from scratch ..

Restarting training from an uninitialized model would have caused us to lose weeks or months of training progress."

OpenAI, et al. "Solving rubik's cube with a robot hand." *arXiv preprint arXiv:1910.07113* 10 (2019).

Deep RL is expensive!



Alphastar: Achieves grandmaster level in Starcraft

- Replication would cost > $1,000,000.
- Excludes most researchers outside resource-rich labs.

Vinyals, Oriol, et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning." *Nature* 575.7782 (2019): 350-354.

Reincarnating RL: An alternative workflow

Reincarnating RL: An alternative workflow

"Prior computational work, such as learned network weights and policies, should be maximally leveraged."

Reincarnating RL: What’s different?

- Lots of related work on imitation + RL, offline RL, transfer, LfD and so on ..
- Such papers typically don't focus on the incorporating such methods as a part of how we do RL research itself.

What would Reincarnating RL look like?

Let's say you trained an agent A_1 for a long time (e.g., weeks) → Experiment with better algorithms / architectures → Training another agent from scratch

(Tabula Rasa)

What would Reincarnating RL look like?

Let's say you trained an agent A_1 for a long time (e.g., weeks) → Experiment with better algorithms / architectures →

- Training another agent from scratch

  (Tabula Rasa)

- Fine-tuning A_1

- Transferring A_1 to another agent and training that agent further

Reincarnating RL as a research workflow

Benefits of Reincarnating RL?

- More compute and sample-efficient

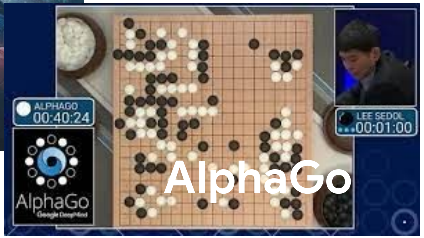
Benefits of Reincarnating RL?

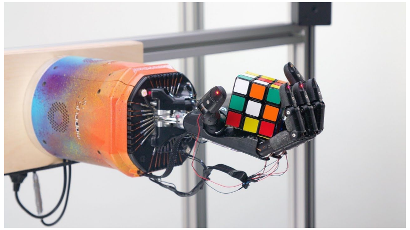
- More compute and sample-efficient
- Allows for continually updating/training agents

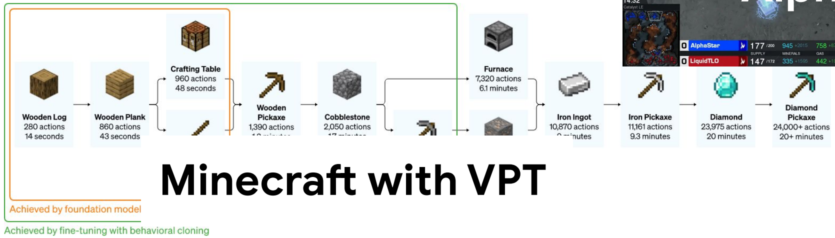
Benefits of Reincarnating RL?

- More compute and sample-efficient
- Allows for continually updating/training agents
- Collaboratively tackling challenging problems

Ad-hoc reincarnation strategies common in large-scale RL







Reincarnating RL ~~common~~ **rare** in typical papers

NEURAL INFORMATION PROCESSING SYSTEMS

ICLR

ICML
International Conference On Machine Learning

Ad-hoc reincarnation strategies common in large-scale RL

Reincarnating RL ~~common~~ **rare** in typical papers

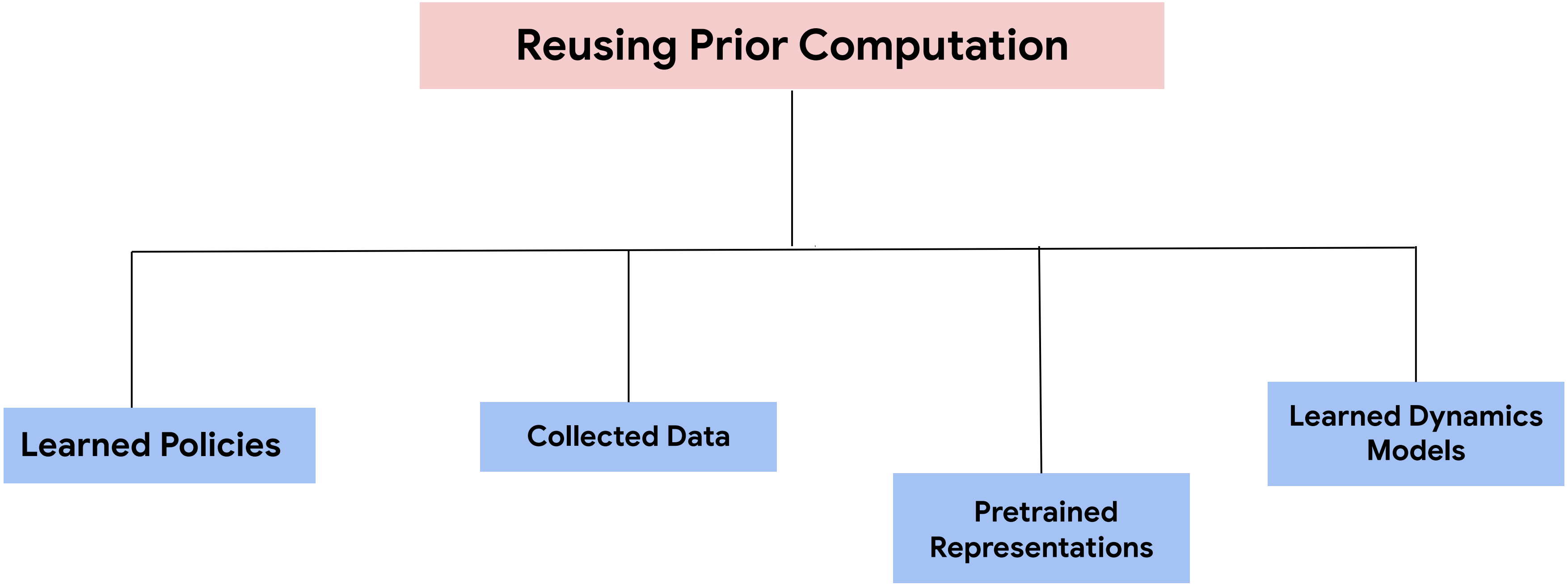
But this doesn't have to be the case!

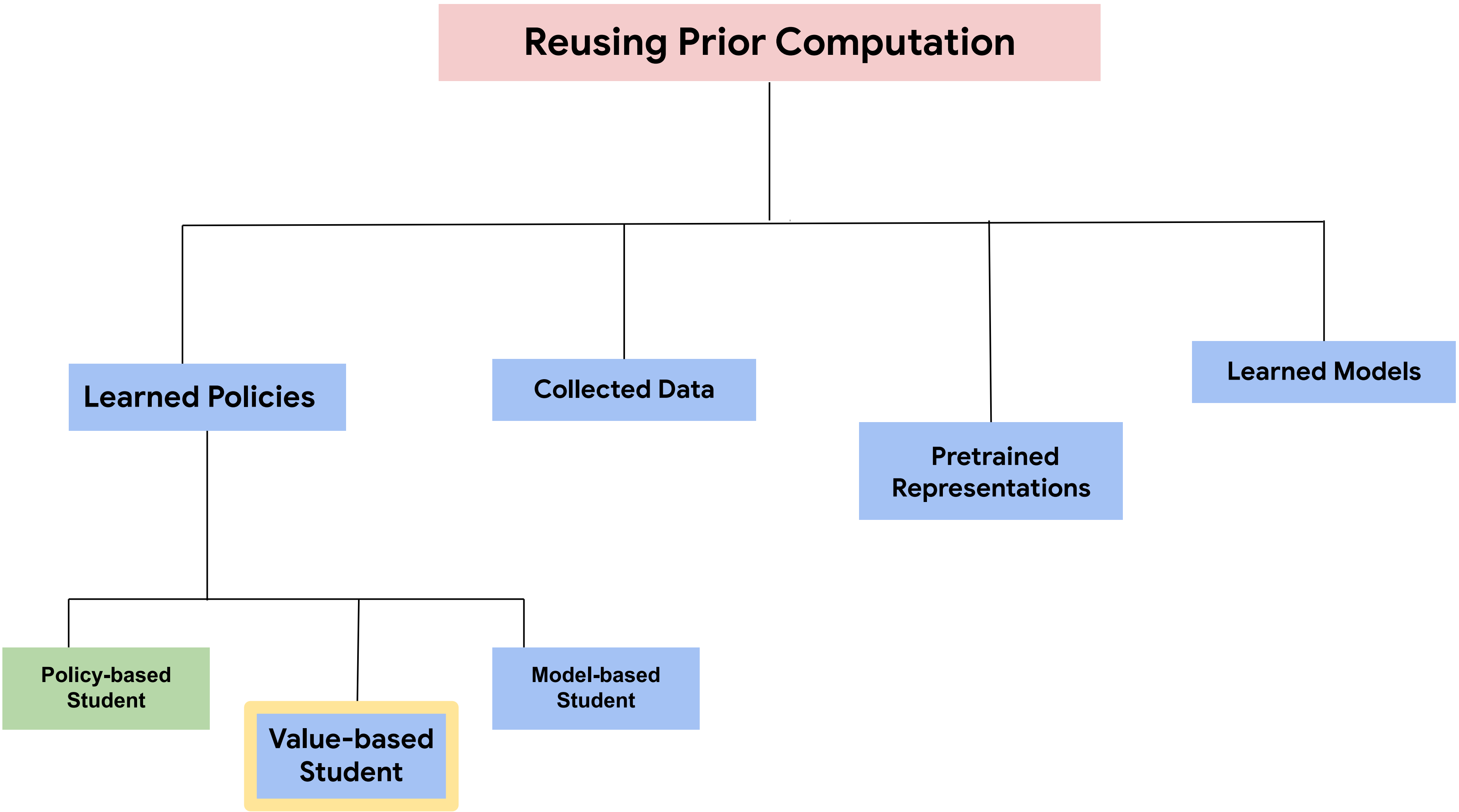
Minecraft with VPT

NEURAL INFORMATION PROCESSING SYSTEMS

ICLR

ICML
International Conference On Machine Learning


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Learned Dynamics Models


Reusing Prior Computation
Learned Policies
Collected Data
Pretrained Representations
Learned Models
Policy-based Student
Value-based Student
Model-based Student

A quick primer on RL

Markov Decision Process (MDP)

S - Set of States

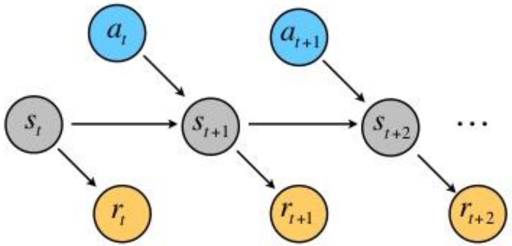
A - Set of Actions

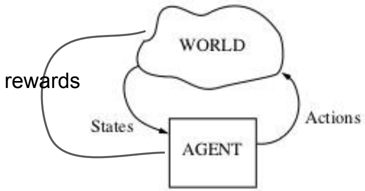
$\Pr(s' \mid a, s)$ - Transitions

α - Starting State Distribution

γ - Discount Factor

$r(s)$ - Reward [or $r(s,a)$]





Goal: $\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t \gamma^t r(s_t, a_t)\right]$

$$s_t \sim P(\cdot \mid s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)$$

A quick primer on RL

How good is a state-action pair?

The Q-function at state s and action a, is the expected cumulative reward from taking action a in state s and then following the policy π. Formally,

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum\nolimits_t \gamma^t R(s_t, a_t) \;\middle|\; s_0 = s, a_0 = a, s_t \sim P(\cdot|s_{t-1}, a_{t-1}), a_t \sim \pi(\cdot|s_t)\right]$$

Bellman Optimality Equation

$$Q^*(s,a) := \max_{\pi} Q^{\pi}(s,a) = \mathbb{E}\left[r(s,a) + \gamma \max_{a'} Q^*(s',a')\right]$$

Solving for the optimal policy

Q-learning: Use a function approximator to estimate the Q-function, *i.e.*

$$Q(s,a;\theta) \approx Q^*(s,a)$$

function parameters (weights)

If the function approximator is a deep neural network => Deep Q-learning!

Case study: Policy to Value Reincarnating RL

$$\pi_\Phi(a|s) \longrightarrow Q_\theta(s, a)$$

Teacher

Student

Transfer an existing policy to a (more) sample-efficient value-based student agent.

Policy to Value Reincarnating RL (PVRL)

$$\pi_{\Phi}(a|s) \longrightarrow Q_{\theta}(s, a)$$

Existing
Teacher Policy

Value-based
Student agent

Desiderata

- **Teacher-agnostic**
  - Student shouldn't be constrained by teacher's architecture and algorithm

Policy to Value Reincarnating RL (PVRL)

$$\pi_{\Phi}(a|s) \longrightarrow Q_{\theta}(s, a)$$

Existing Teacher Policy

Value-based Student agent

Desiderata

- Teacher-agnostic
- **Weaning off teacher**
  - Undesirable to maintain teacher dependency for successive reincarnations

Policy to Value Reincarnating RL (PVRL)

$$\pi_{\Phi}(a|s) \longrightarrow Q_{\theta}(s, a)$$

Existing Teacher Policy

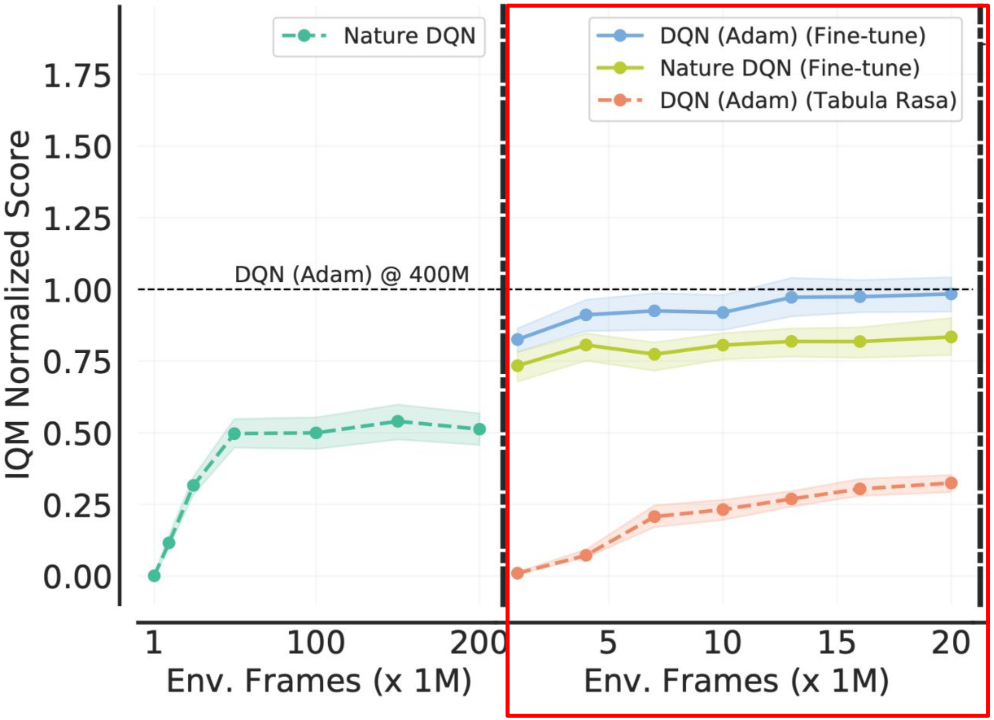
Value-based Student agent

Desiderata

- Teacher-agnostic
- Weaning off teacher
- **Sample Efficient**
    - Reincarnation should be cheaper than training from scratch

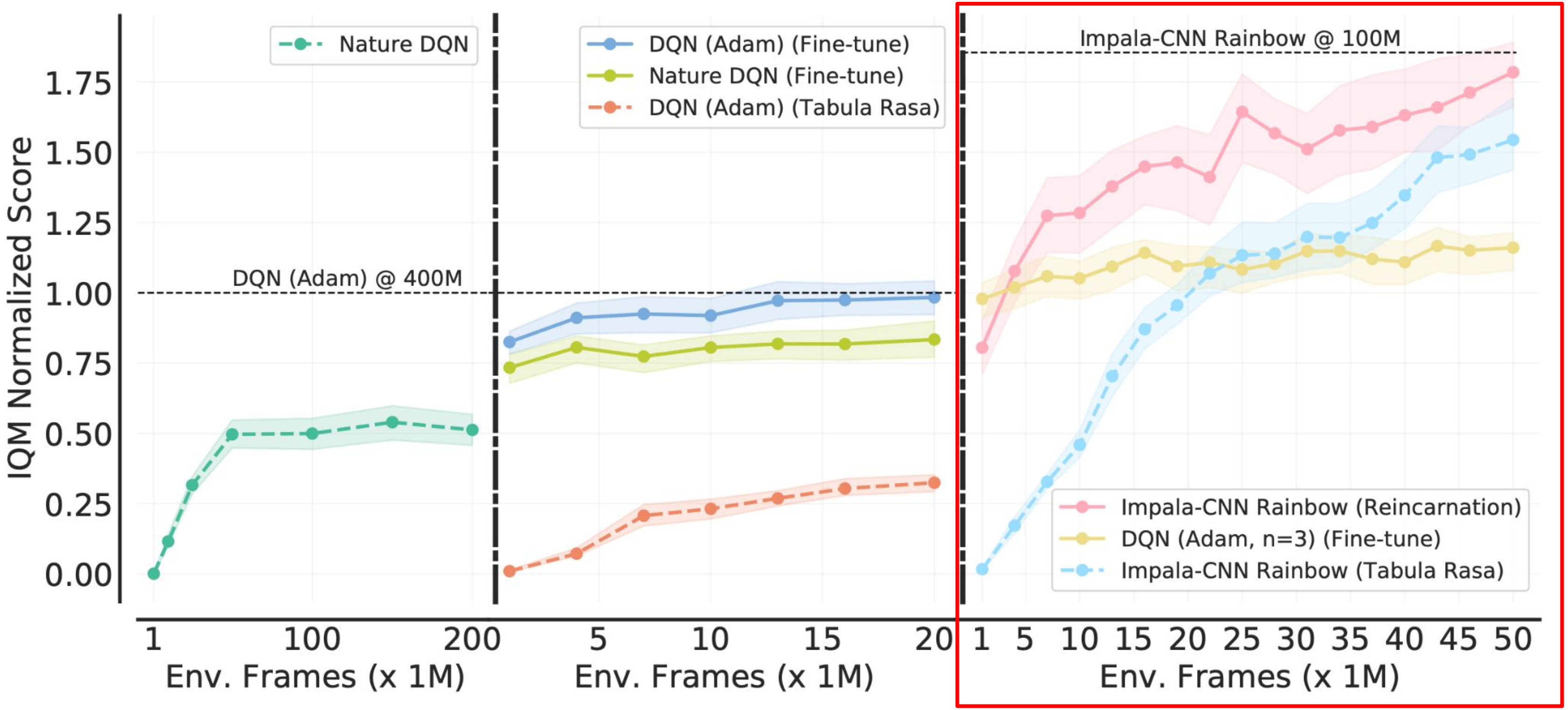
Reincarnating RL as a workflow

Similar results in few hours of training rather than a week!



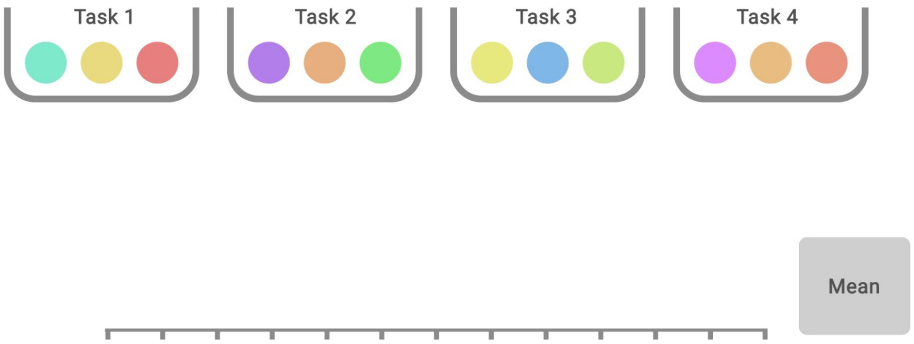
Reincarnating RL as a workflow

Saved 50M frames or 1 day of GPU training!



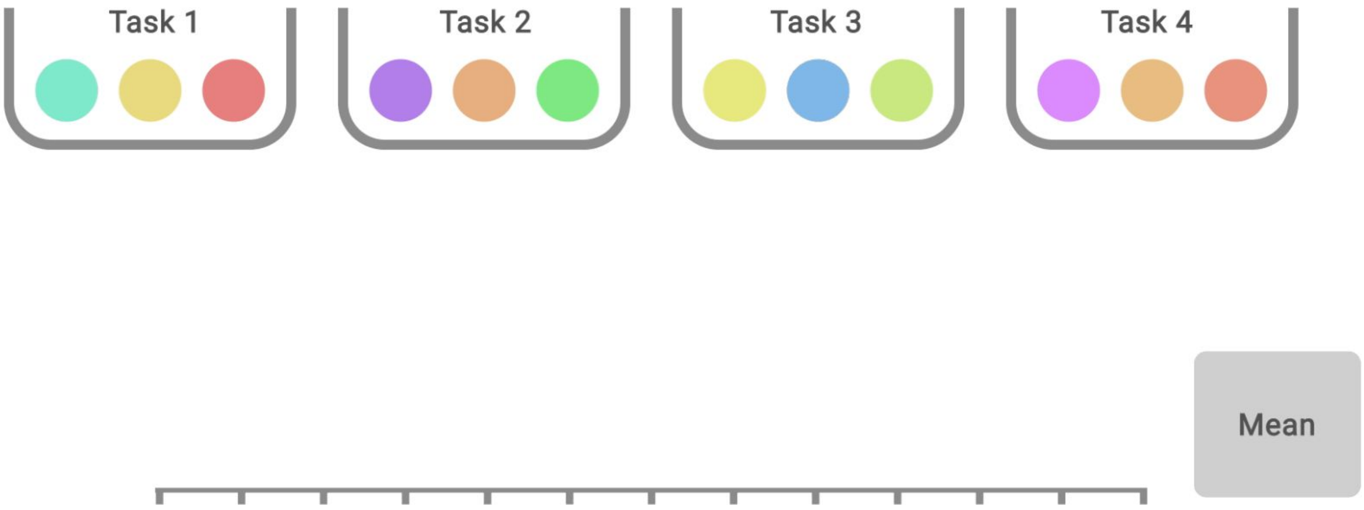
PVRL: Experimental Setup

- Interactive teacher policy: DQN trained for 400M frames (**7 days**)
  - Also assume access to replay data of the teacher
- Transfer a student DQN using 10M frames (a few hours)
- 10 Atari games with sticky actions (for stochasticity)
- Evaluation: Interquartile Mean [1]



[1] For more details, see **Deep RL at the Edge of the Statistical Precipice.** NeurIPS 2021 (Best Paper).

A note about evaluation: Interquartile Mean



IQM discards the lowest 25% and highest 25% of the combined scores (colored balls) and computes the mean of the remaining 50% scores.

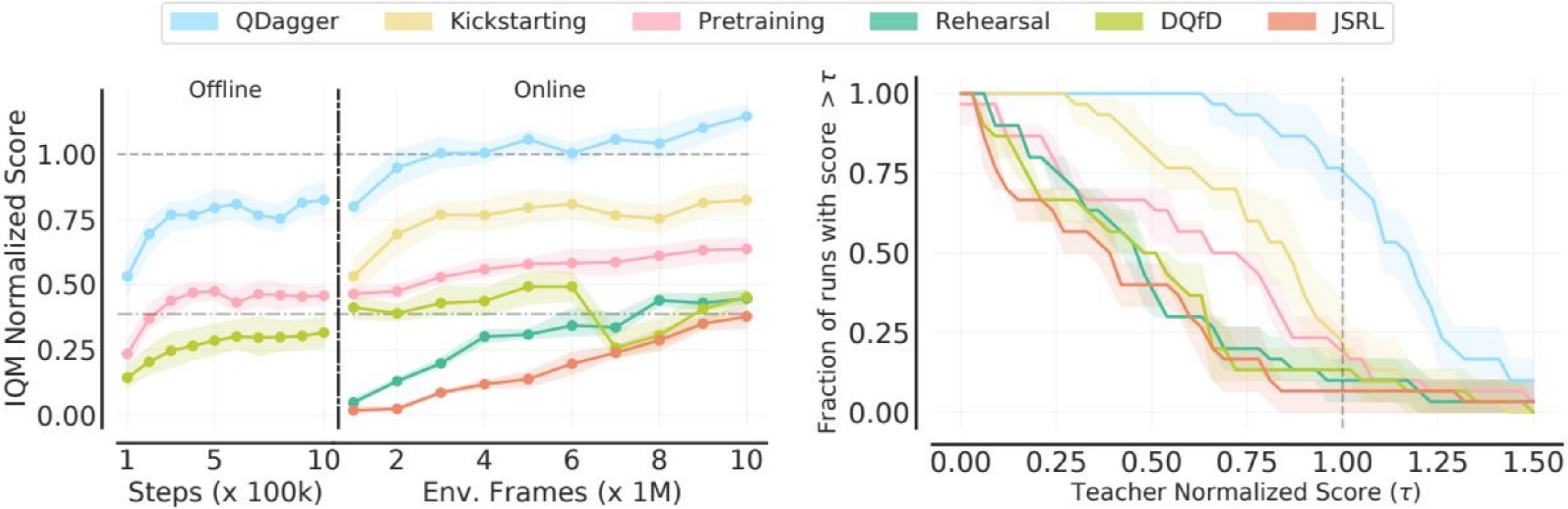
For more details, see **Deep RL at the Edge of the Statistical Precipice.** NeurIPS 2021 (Outstanding Paper).

PVRL: Closely-related approaches

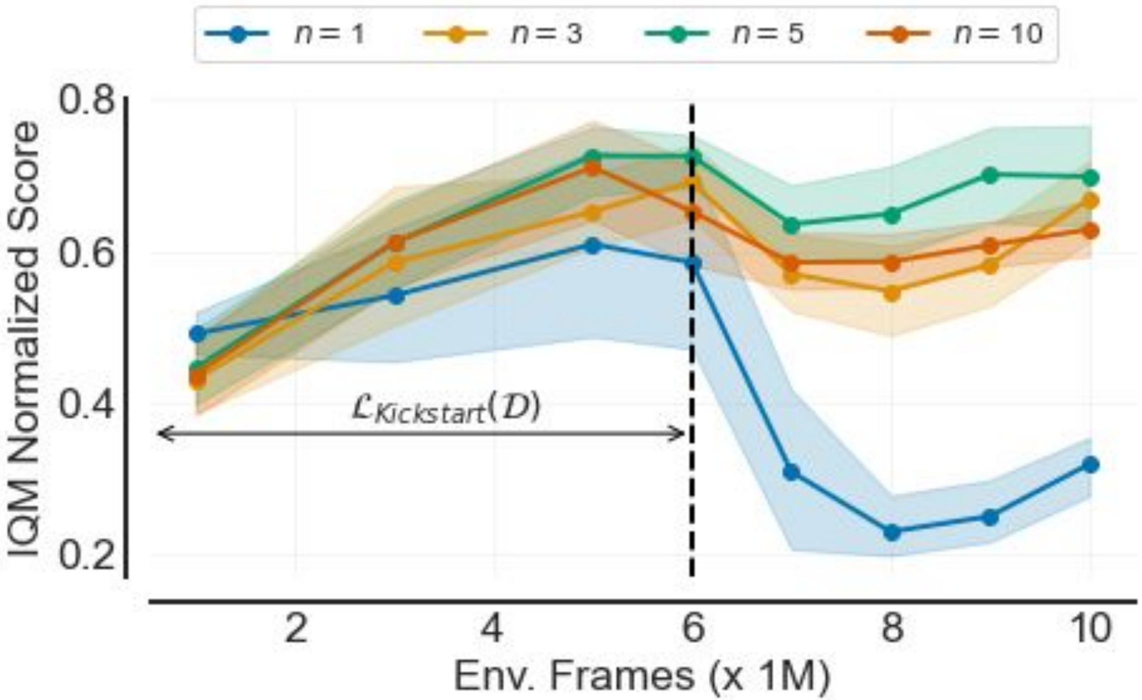
Adapting existing approaches:

- **Rehearsal:** Replaying Teacher Samples
- **Pretraining:** Offline RL on Teacher Data
- **Kickstarting:** On-policy Distillation + Q-learning
- **DQfD:** Learning from teacher demonstrations
- **JSRL**: Improving exploration using teacher

PVRL results on ALE



Kickstarting (On-policy Distillation + RL)



QDagger: A simple PVRL baseline

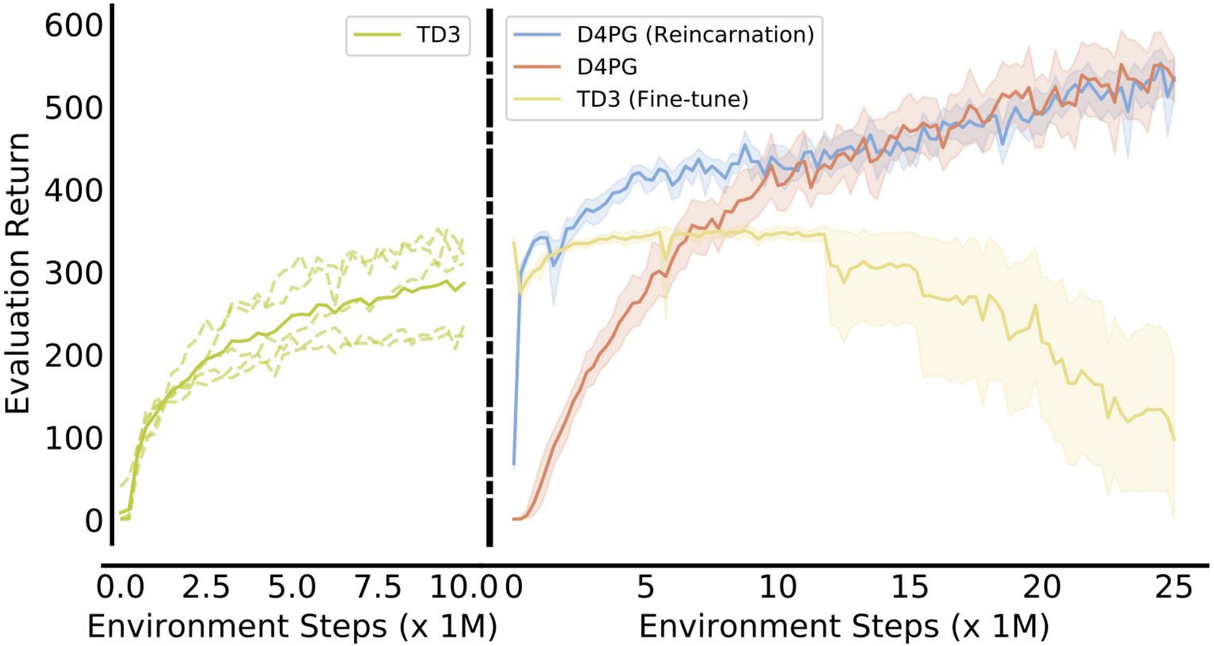
$$\mathcal{L}_{QDagger}(\mathcal{D}) = \underbrace{\mathcal{L}_{TD}(\mathcal{D})}_{\text{Q-learning loss}} + \lambda_t \mathbb{E}_{s\sim\mathcal{D}}\Big[\sum_a \underbrace{\pi_T(a|s)\log\pi(a|s)}_{\text{On-policy distillation}}\Big]$$

Decaying coefficient to wean off the teacher.

Combine Q-learning with Dagger. Phases:

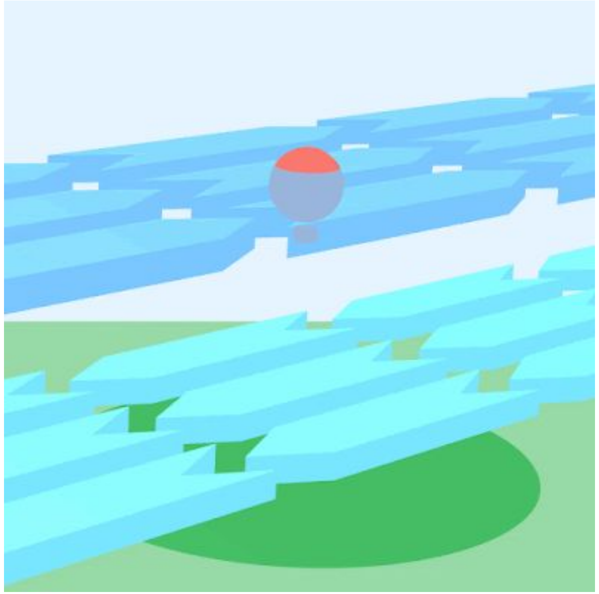
- (Offline) Pretrain on Teacher data
- (Online) Train on self-collected data.

Tackling a hard control task: Humanoid run



Saved 10M frames (10-12 hours on a V100)

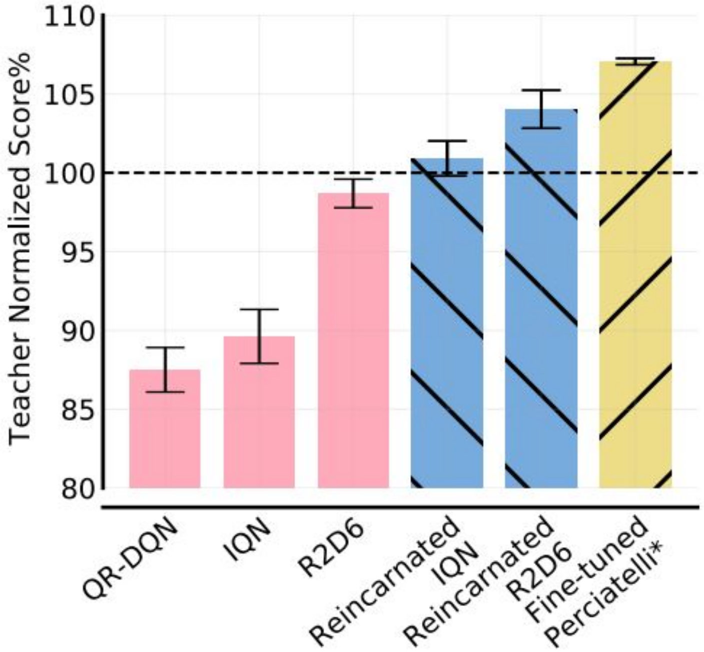
Making progress on BLE



- Access to the Perciatelli QR-DQN agent trained for a month.
- Given access to finite compute (10-12 hours on a TPU-v2), how much progress can be made?

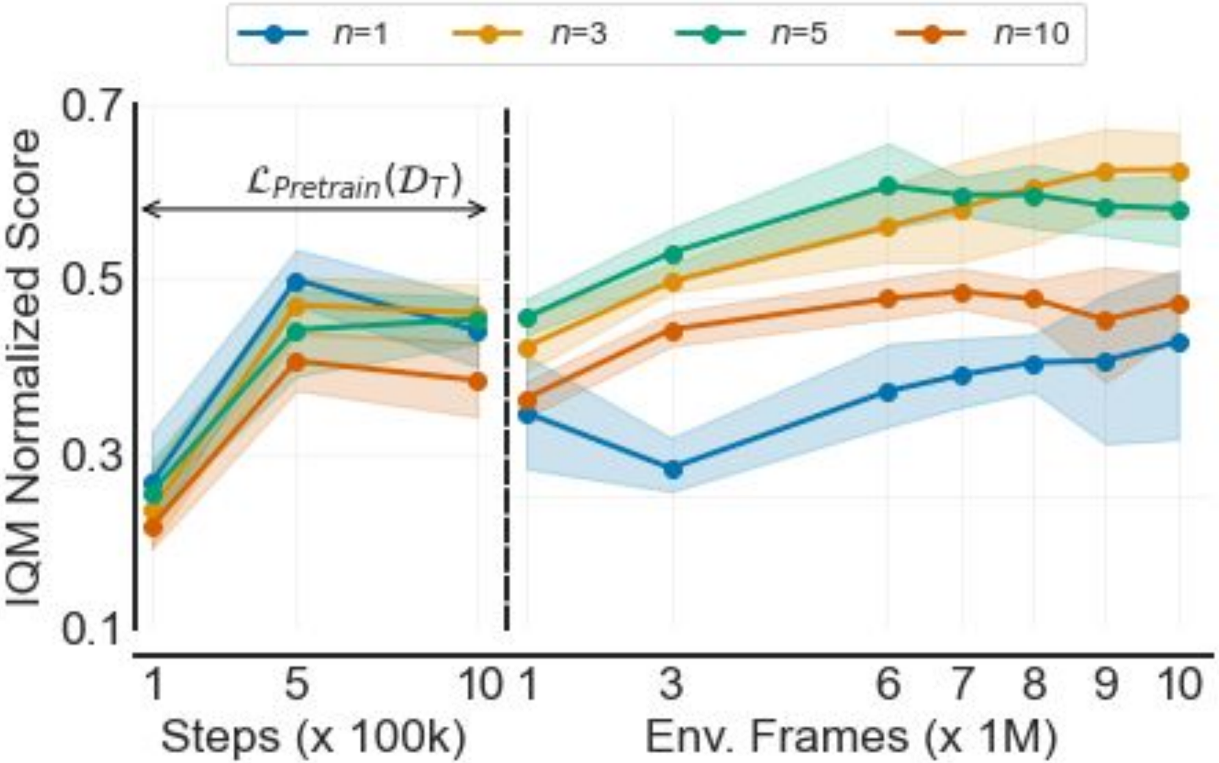
[1] Bellemare, Marc G., et al. "Autonomous navigation of stratospheric balloons using reinforcement learning." *Nature* 588.7836 (2020): 77-82.
[2] The Balloon Learning Environment. https://ai.googleblog.com/2022/02/the-balloon-learning-environment.html

Making progress on BLE

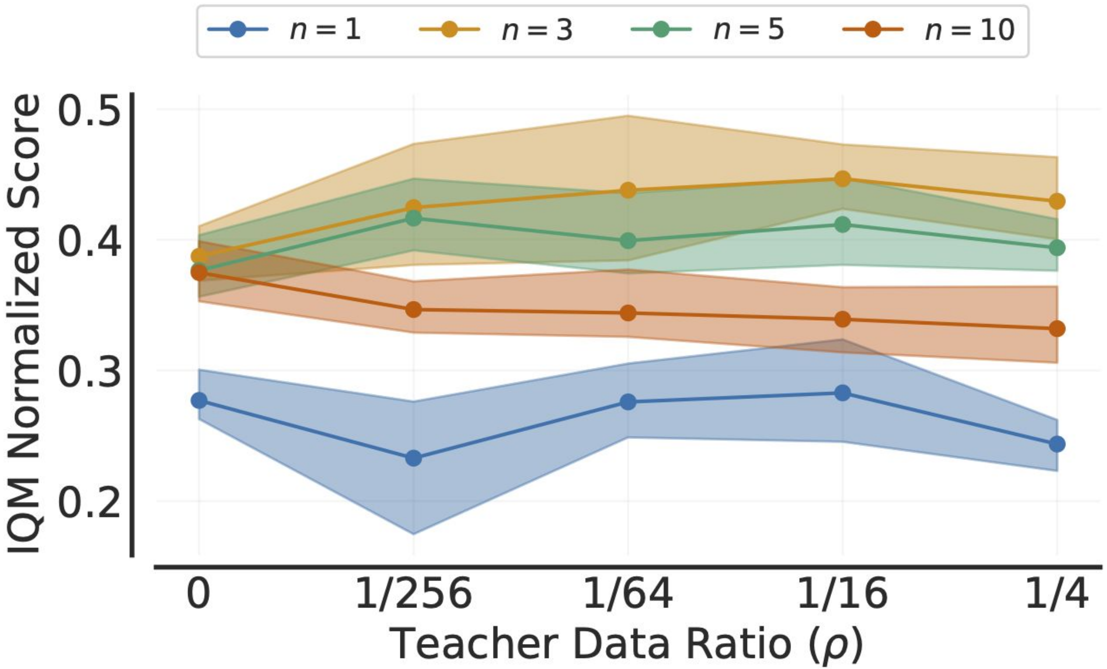


[1] Bellemare, Marc G., et al. "Autonomous navigation of stratospheric balloons using reinforcement learning." *Nature* 588.7836 (2020): 77-82.
[2] The Balloon Learning Environment. https://ai.googleblog.com/2022/02/the-balloon-learning-environment.html

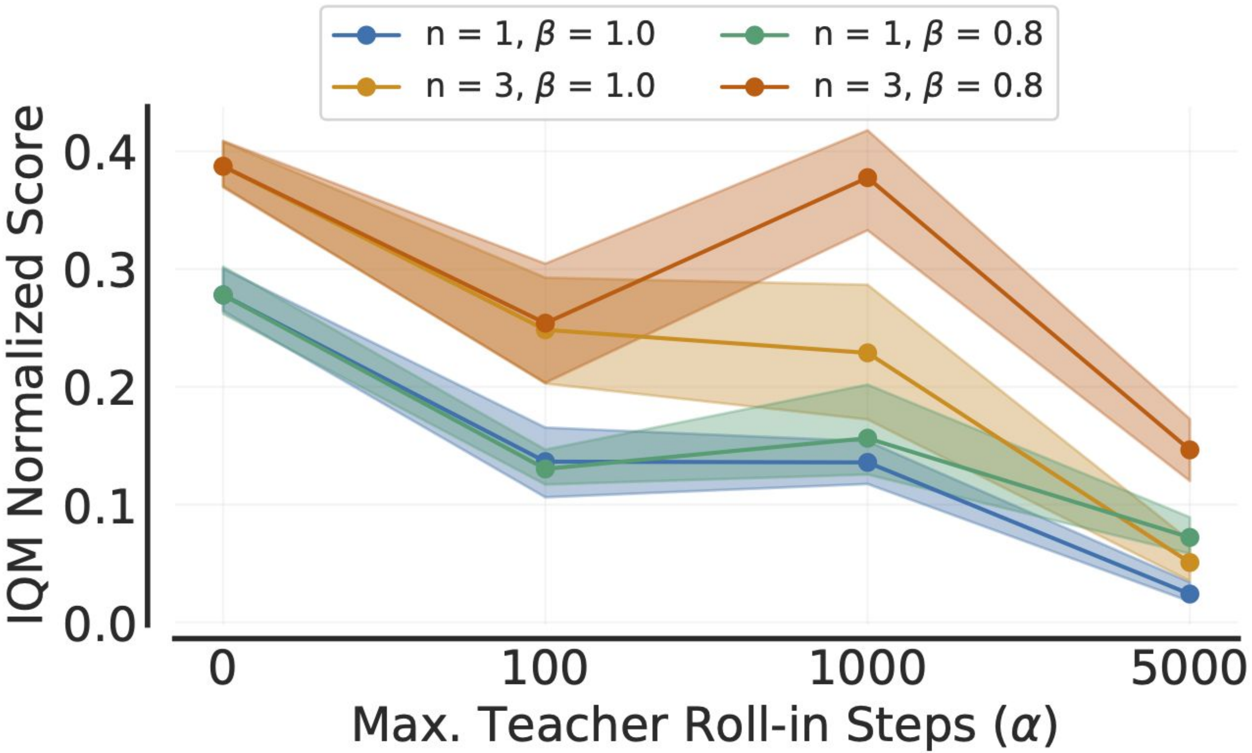
Offline Pretraining



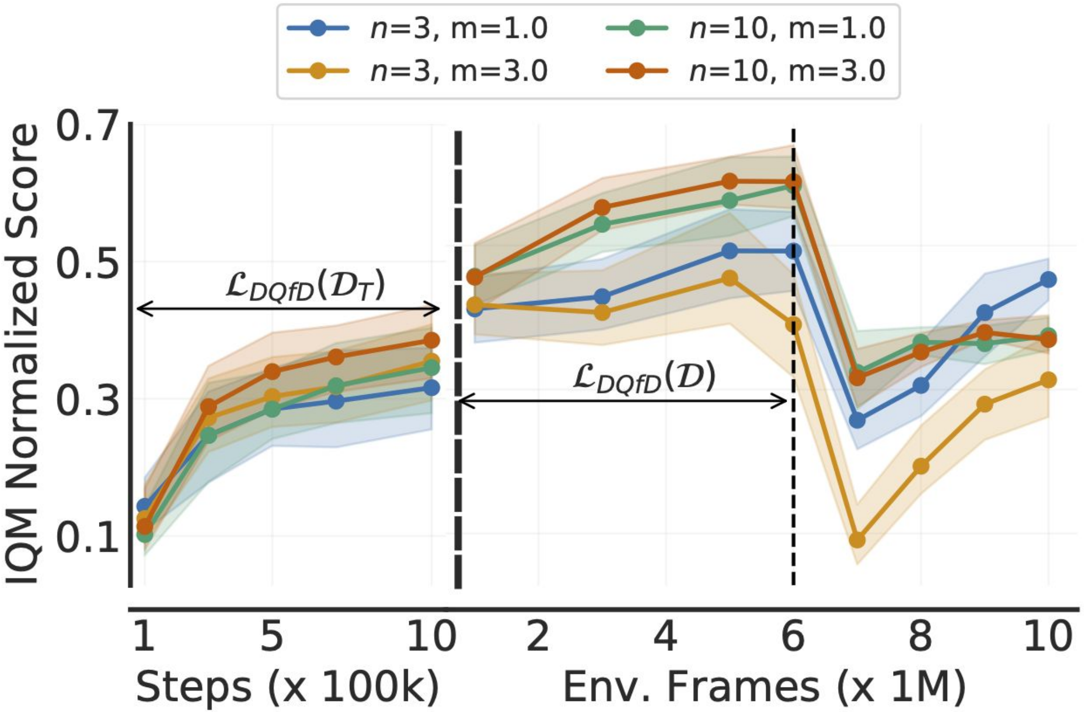
Teacher Rehearsal



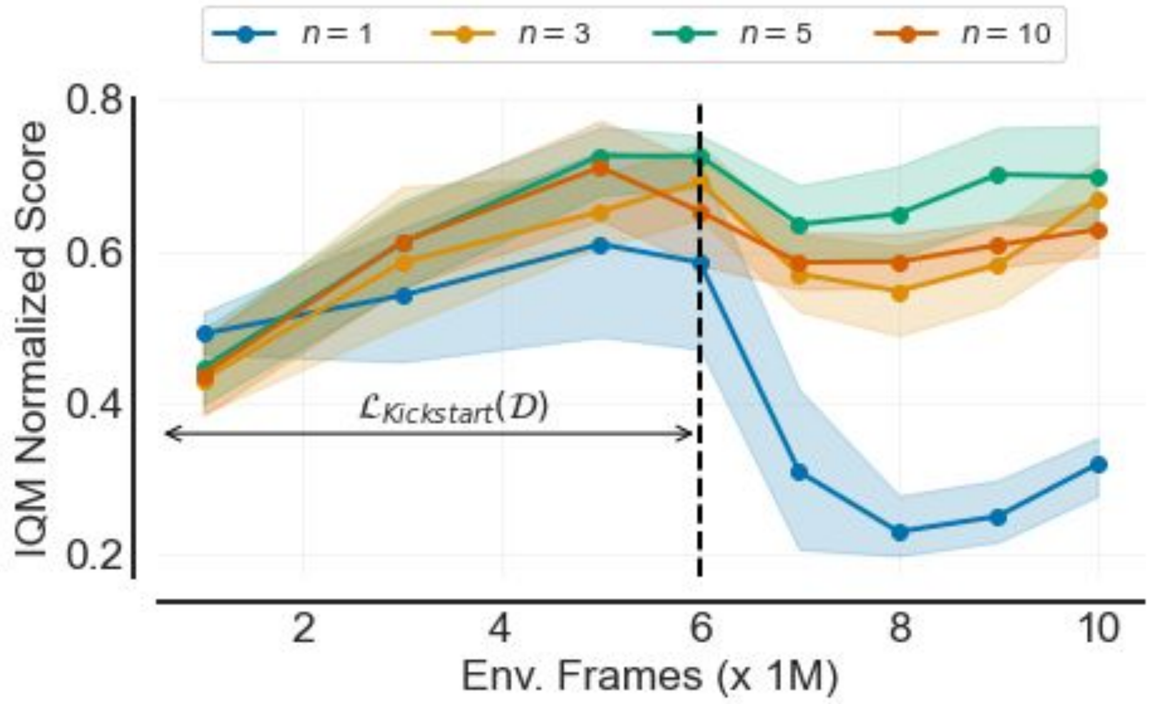
Jump-Start Reinforcement Learning



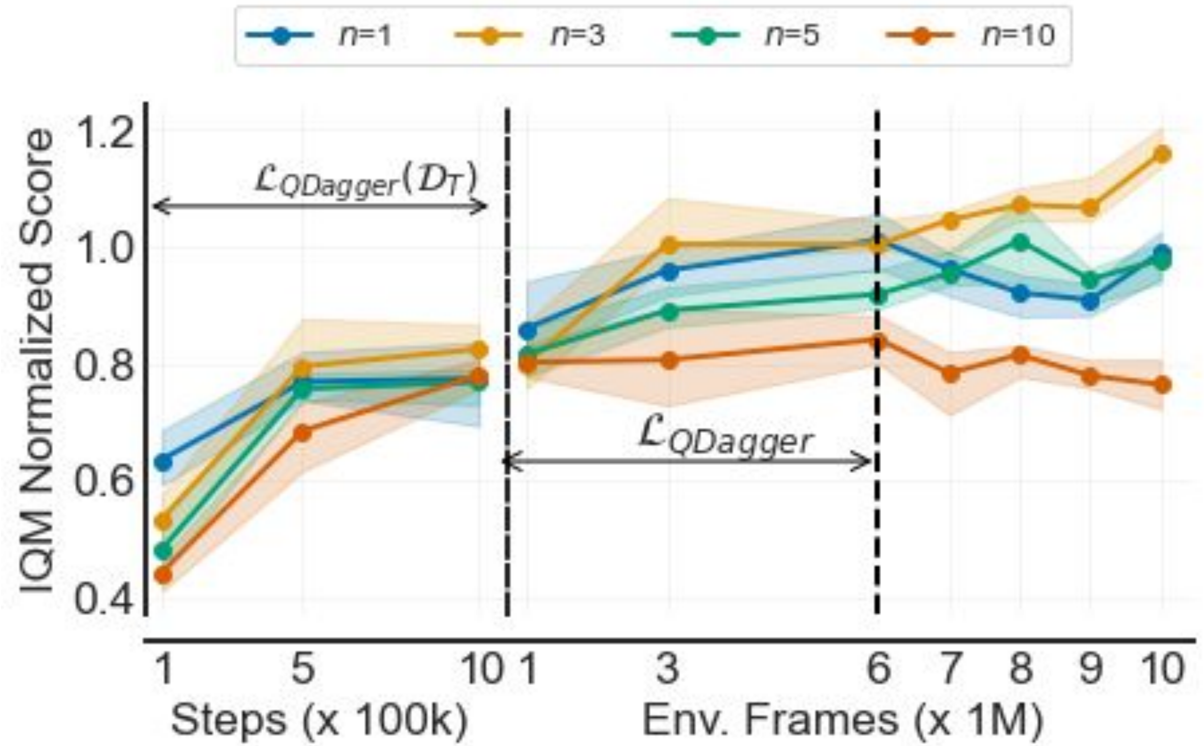
Deep Q-learning from Demonstrations



Kickstarting (On-policy Distillation + RL)

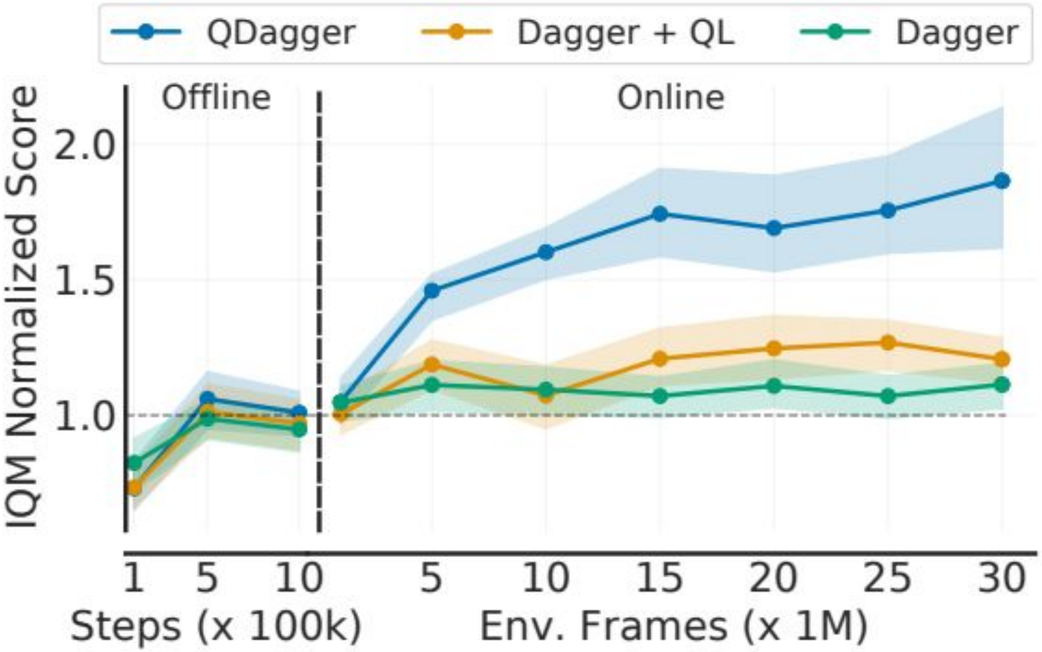


Our naive method: Q-Dagger (Dagger + Q-learning)

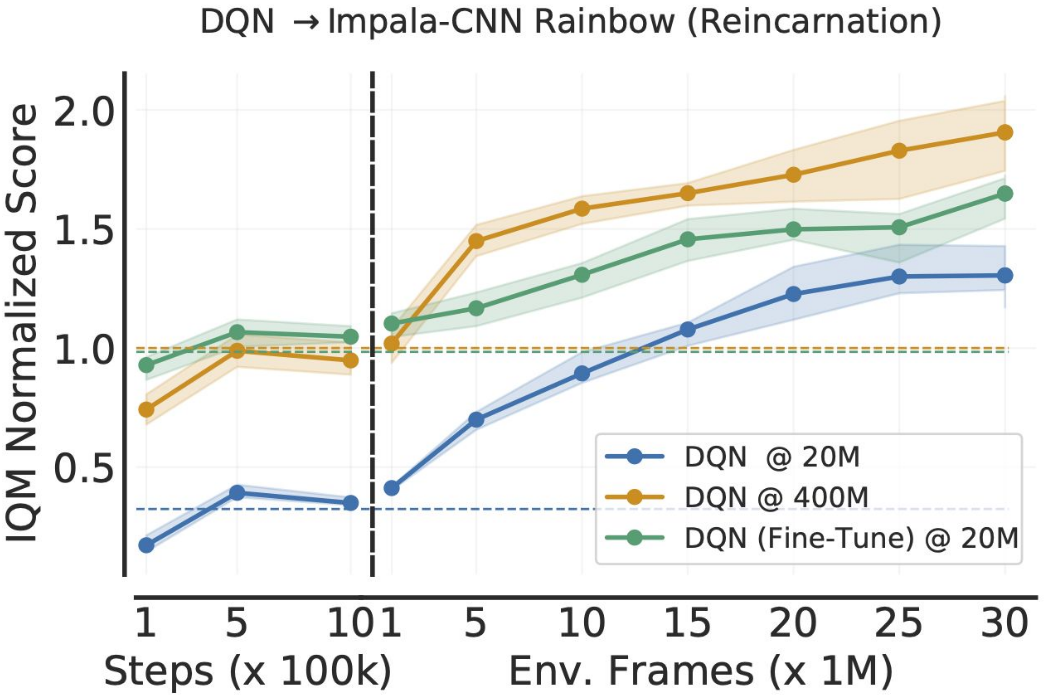


Considerations in Reincarnating RL

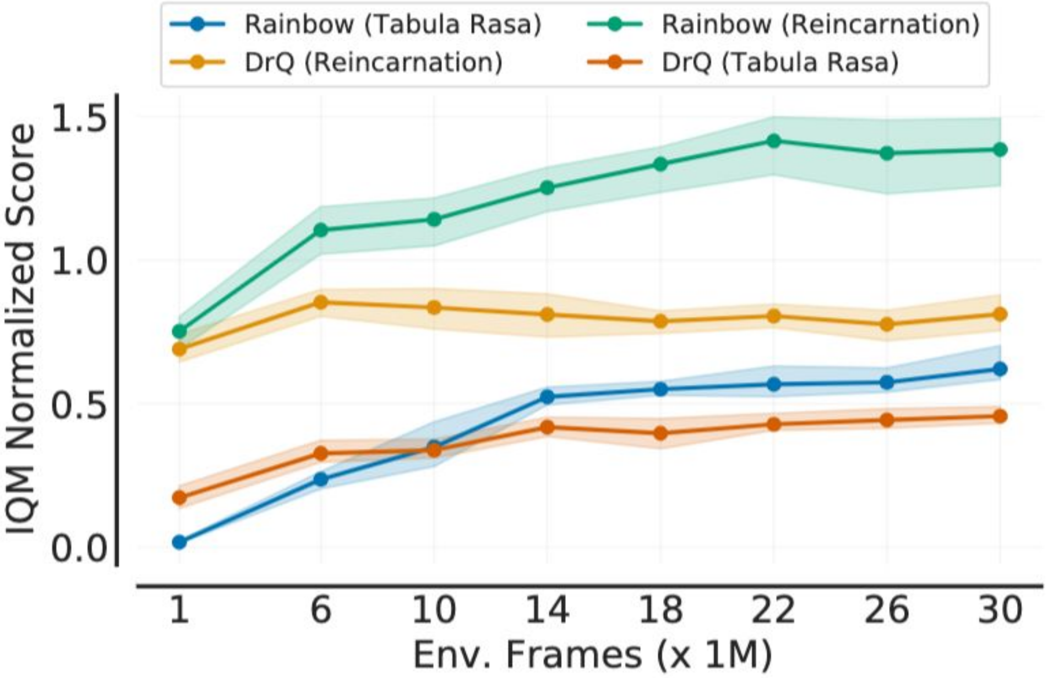
Reincarnation vs Distillation



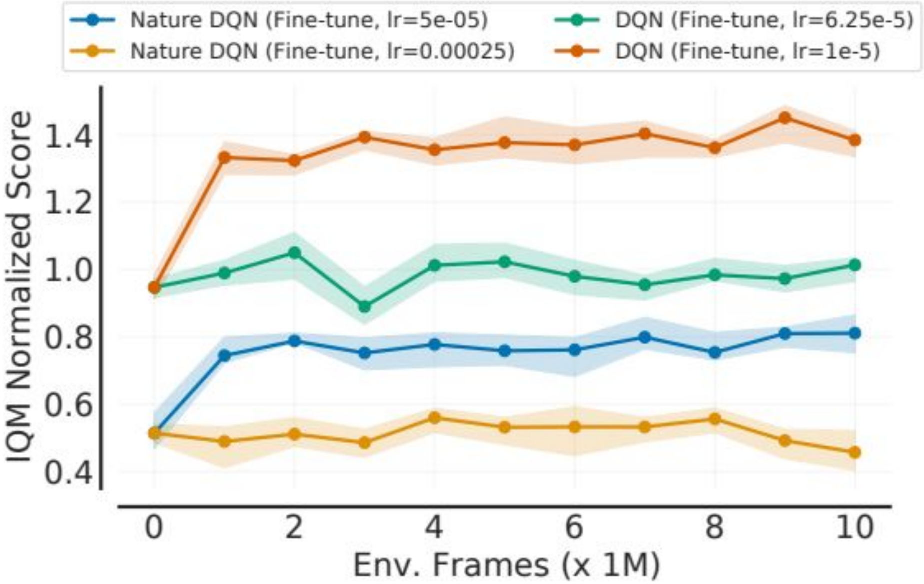
Dependence of Prior Computation



Benchmarking Differences with Tabula Rasa



Fine-tuning for Reincarnation



"If I have seen further than others, it is by standing upon the shoulders of giants."

– Sir Isaac Newton